AFRL-RX-WP-TP-2009-4129

DYNAMIC CHANNEL ALLOCATION IN WIRELESS NETWORKS USING LEARNING AUTOMATA (PREPRINT)

Behdis Eslamnour, Maciej Zawodniok, and Jagannathan Sarangapani

Missouri University of Science and Technology

MARCH 2009

AIR FORCE RESEARCH LABORATORY
MATERIALS AND MANUFACTURING DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7750
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
March 2009	Journal Article Preprint	

4. TITLE AND SUBTITLE
DYNAMIC CHANNEL ALLOCATION IN WIRELESS NETWORKS USING LEARNING AUTOMATA (PREPRINT)

5a. CONTRACT NUMBER
FA8650-04-C-5704

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
78011F

6. AUTHOR(S)
Behdis Eslamnour, Maciej Zawodniok, and Jagannathan Sarangapani

5d. PROJECT NUMBER
2865

5e. TASK NUMBER
25

5f. WORK UNIT NUMBER
25100000

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Missouri University of Science and Technology
Mechanical and Aerospace Engineering Department
400 W. 13th Street
Rolla, MO 65409-0050

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Materials and Manufacturing Directorate
Wright-Patterson Air Force Base, OH 45433-7750
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RXLMP

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RX-WP-TP-2009-4129

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Journal article submitted to the *IEEE Transactions on Wireless Communications*. PAO Case Number: 88 ABW-2009-0726; Clearance Date: 26 Feb 2009. Paper contains color.

14. ABSTRACT
Single channel based wireless networks have limited bandwidth and throughput and the bandwidth utilization decreases due to congestion and interference from other sources. In order to increase the throughput, transmission in multiple channels is considered as an option. In this paper, we propose a distributed dynamic channel allocation scheme using adaptive learning automata for wireless networks whose nodes are equipped with single radio interfaces. The proposed schemes, Adaptive Pursuit Reward-Inaction, Adaptive Pursuit Reward-Penalty, and Adaptive Pursuit Reward-Only, run periodically on the nodes, and adaptively find the suitable channel allocation in order to attain a desired performance. A novel performance index, which takes into account the throughput and the energy consumption, is considered. The proposed learning scheme is adaptive in the sense of updating rule. The update value of the probabilities in the each step is a function of the error in the performance index.

15. SUBJECT TERMS
Adaptive reward-inaction, Channel Allocation, Learning Automata, Wireless Networks

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
Unclassified	Unclassified	Unclassified	SAR	30	Todd J. Turner
					19b. TELEPHONE NUMBER *(Include Area Code)* N/A

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Dynamic Channel Allocation in Wireless Networks using Learning Automata

Behdis Eslamnour, *Student Member IEEE*, S. Jagannathan *Sr. Member IEEE, and* Maciej Zawodniok, *Member IEEE*

***Abstract*—** Single channel based wireless networks have limited bandwidth and throughput and the bandwidth utilization decreases due to congestion and interference from other sources. In order to increase the throughput, transmission in multiple channels is considered as an option. In this paper, we propose a distributed dynamic channel allocation scheme using adaptive learning automata for wireless networks whose nodes are equipped with single radio interfaces. The proposed schemes, Adaptive Pursuit Reward-Inaction, Adaptive Pursuit Reward-Penalty, and Adaptive Pursuit Reward-Only, run periodically on the nodes, and adaptively find the suitable channel allocation in order to attain a desired performance. A novel performance index, which takes into account the throughput and the energy consumption, is considered. The proposed learning scheme is adaptive in the sense of updating rule. The update value of the probabilities in the each step is a function of the error in the performance index. Comparing the three schemes in simulation environment, it was shown that the Adaptive Pursuit Reward-Only scheme guarantees updating the probability of the channel selection for all the links – even the links that their current channel allocations do not provide a satisfactory performance, hence reducing the frequent channel switching on the links that cannot achieve the desired performance on any of the channels.

***Index Terms*—** Adaptive reward-inaction, Channel Allocation, Learning Automata, Wireless Networks.

I. Nomenclature

Symbol	Definition	Symbol	Definition
N	number of channels	$J_i^j(k)$	percentage of successful transmissions

This work was supported in part by the AFRL Contract and Intelligent Systems Center.

Behdis Eslamnour, Maciej Zawodniok, and Jagannathan Sarangapani are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology (formerly University of Missouri-Rolla), Rolla, MO 65409 USA (email: {ben88, mjzx9c, sarangap}@mst.edu).

C	set of available channels, $C = \{c_1, c_2, \ldots, c_N\}$	$L_i^j(k)$	number of times that channel j was selected for node i from time 0 till k
$p_i^j(k)$	probability of node i selecting channel j at time k, $\sum_{j=1}^{N} p_i^j(k) = 1$	$\hat{E}_i^j(k)$	average estimated consumed energy over a window of M $\hat{E}_i^j(k) = \frac{1}{M} \sum_{n=L_i^j(k)-M+1}^{L_i^j(k)} e_i^j(k)$
$\mathbf{P}_i(k)$	probability vector of node i selecting the N channels, $\mathbf{P}_i(k)=[p_i^1(k), p_i^2(k), \ldots, p_i^N(k)]$	ϕ^*	desired performance (joules/packet)-1 $\phi^* = \left(\frac{H}{E}\right)_{desired}$
$\beta_i^j(k)$	environment response for selecting channel j by node i at time k $\begin{cases} \text{if } \beta_i^j(k) = 0, \text{ the autumaton will be rewarded} \\ \text{if } \beta_i^j(k) = 1, \text{ the automaton will not be rewarded} \end{cases}$	$\hat{\phi}_i^j(k)$	estimated performance of channel j for node i at time k $\hat{\phi}_i^j(k) = \frac{\hat{H}_i^j(k)}{\hat{E}_i^j(k)}$
$\hat{H}_i^j(k)$	average estimated throughput over a window of M $\hat{H}_i^j(k) = \frac{1}{M} \sum_{n=L_i^j(k)-M+1}^{L_i^j(k)} J_i^j(n)$	$\hat{m}_i$	index of the channel that provides the maximum estimated performance at time k $\hat{m}_i = \arg \max_j \hat{\phi}_i^j(k)$

II. INTRODUCTION

IT is widely believed that the wireless networks are being limited by the lack of the available spectrum, and at the same time the spectrum is not efficiently utilized. Spectrum utilization can be improved using spatial techniques, frequency, modulation techniques, *etc*. As a consequence, newer concepts such as software-defined radios and cognitive radios were made possible [1]. While the cognitive radios are not limited to spatial and temporal spectrum utilization, the spatial channel reuse approach in wireless networks has been vastly investigated [2]-[7].

The bulk of the research on multiple channel allocation is notably done for mesh networks [3],[7], WLANs with infrastructure [4], cellular networks [8] and cognitive radio networks [5]. The multi-channel allocation problem has been investigated for the networks in which the nodes are equipped with either multiple-radio interface [7]-[10] or single-radio interface [2],[4],[11]-[13]. In the single-radio approach, the radios switch between the channels frequently in order to minimize interference and collision between the simultaneous transmissions in the same communication range. Usually in this approach, all the nodes periodically switch to a common channel for channel co-ordination, and then switch to different data channels to conduct the simultaneous transmissions. Therefore the switching delay (80-100 μs [2]) becomes one of the overheads increasing the network end-to-end delay. Additionally, synchronization is required in these schemes.

In the networks with infrastructure and access points [4], the channel co-ordination signals are exchanged through the wired distribution system connecting the access points. This practically eliminates the need for periodically switching to a common channel. In the case of multiple-radio interface approach, usually one interface is dedicated to the control signals, and the remaining channels are allocated for simultaneous transmission of data thus increasing temporal and spatial spectrum utilization and not requiring synchronization. Further, utilizing multiple radios reduces the need for frequent channel switching, and hence the switching overhead is significantly less than that in the single-radio approach. However, the cost of additional radios and their energy consumption must be taken into account.

By contrast, in this paper, we propose a distributed dynamic channel allocation scheme for wireless networks and in particular wireless sensor networks particular wireless sensor networks whose nodes are equipped with single radio interface due to their low cost requirement. Therefore, synchronization is required in this scheme. The periodic nature of this algorithm makes it dynamic and enables the channel allocation to adapt to the topographic changes, possible loss of some channels, mobility of the nodes, and the traffic flow changes. The adaptive pursuit learning algorithm runs periodically on the nodes, and adaptively finds the optimum channel allocation that provides the desired performance (or closest to the desired performance). Unlike the linear and nonlinear schemes in which the reward and penalty values were functions of the probabilities, we examine an adaptive updating scheme in which the reward and penalty values are functions of the error between the desired and the estimated performance of the current channel allocation. By selecting realistic desired performance metric, the convergence of the algorithm is guaranteed.

In Section III, the methodology and algorithms are presented. Simulation results and discussions are provided in Section IV. Section Vconcludes the paper. Proof of convergence of the algorithm is presented in Appendix A.

III. METHODOLOGY AND ALGORITHM

A. Methodology

In the proposed algorithm, the nodes periodically switch between the control stage, T_c, and data transmission stage, T_d (Figure 1). Each data transmission period, T_d, is comprised of the individual time slots, T_s. As an initial assumption, we consider peer-to-peer networks in which all nodes are equipped with a single radio. We also assume that routes have been established by a proactive routing protocol such as optimal link state routing (OLSR) [17] or optimal energy delay routing (OEDR) [18]. During T_c, all nodes are on one common channel to communicate the control signals. It is possible that one or more of the channels get highly affected by external interference and the network would lose these channels temporarily or permanently.

In order to maintain the network connectivity in the sense of exchanging the control signals, we propose having a unique sequence of all the channels. In the event of a loss of a control channel, the nodes would try the next channel in the sequence as the control channel during T_c. The control signal carries schedule of the time slots for the links in the subsequent data transmission period. During the time scheduling, groups of non-intersecting links are scheduled for each T_s time slot. Also broadcast communications and route discovery are performed during T_c period. After the T_c stage, the data transmission stage, T_d, begins. During each T_s time slot of T_d, channels are allocated to the links previously assigned to the T_s. The channel allocation algorithm is an iterative algorithm during which the channel allocation is refined. Due to the iterative nature of the algorithm, each T_s is divided into smaller time slots, T_{mini}, separated by T_g – guard bands. The probabilities and parameters of the channel allocation algorithm are updated for each link from one T_{mini} to the next.

By periodically repeating the T_c and T_d stages, the channel allocation becomes dynamic. In addition, the network can adapt to the topographic changes, mobility of the nodes, and the changes in the traffic flow. Also in the event of control channel, C_c, loss the next channel in the sequence will be used as the control channel. It must be noted that this sequence is a common knowledge among all the nodes in the network. Any eligible external node that tries to join the network would send out join-request signals periodically and listen in the intervals. It would be able to join the network during one of the T_c periods, and obtain the sequence and other necessary information about the network.

We also propose using the control channel as one of the available channels for data transmission during the T_d period. By utilizing this additional channel during T_d instead of dedicating it to the control signals and using it only during T_c, the spectrum utilization can be increased.

B. Algorithm

During each T_s, the learning algorithm is run on each transmitter node, *i*, separately. We first use the Adaptive Pursuit Reward-Inaction (PRI) which is an extended version of Distributed PRI [14],[15]. Unlike the DPRI, in the Adaptive PRI scheme the update value, $\theta(k)$, of the probabilities is not a constant anymore. The update value of the probability is now a function of the error, $\Delta(k)$, of the performance metric. We chose DPRI algorithm because of the faster convergence provided by it [14]. The Adaptive PRI algorithm is presented in Section B.1. However, it appears that depending on the conditions that determine whether the environment response is satisfactory or unsatisfactory, the channel allocation on some links might always result unsatisfactory response. This would result in 'left-out' links, whose channel selection probabilities are not updated due to the 'reward' property of the algorithm.

In order to eliminate this issue, we proposed and examined the Adaptive Pursuit Reward-Penalty (PRP) learning scheme. The 'reward' behavior of this scheme is the same as the Adaptive PRI. On the other hand, in the case of unsatisfactory environment response for a channel selection, the probability of selecting that channel (if that channel is not the channel with the highest performance among the channels) is decreased, and the probabilities of selecting the other channels are increased. The algorithm is presented in Section B.2. Although this scheme eliminates the 'left-out' links problem, it has a rather slower convergence because of increasing the probabilities of some of the non-optimal channels in the 'penalty' scheme.

In the third effort, we proposed and examined using an Adaptive Pursuit Reward-Only (PRO) learning algorithm. In this algorithm we still use a desired value of the performance for determining the magnitude of the update step in the probabilities, but we no longer use the concept of ‘satisfactory’ or ‘unsatisfactory’ environment response. In other words, the Adaptive PRO is the same as the Adaptive PRI, but the probabilities are guaranteed to be updated in a ‘pursuit reward’ manner at each iteration.

The performance metric of the network used in this paper was defined as

$$\phi^* = \left(\frac{H}{E}\right)_{desired} \quad (1)$$

where H is the desired percentage of the successful transmissions and E refers to the desired consumed energy per one successful packet transmission. By this definition, the unit of the performance metric ϕ^* becomes packets/joule. Therefore, by selecting a realistic desired performance metric, the objective is to find the optimum channel allocation that provides a higher performance in terms of throughput defined in terms of a target value. A large value of ϕ^* indicates successful transmission of more packets. Hence, this performance metric covers both the throughput and the energy efficiency of the network.

B.1. The Adaptive Pursuit Reward-Inaction Algorithm

The steps of the Adaptive PRI, which runs on each individual link, are summarized as:

1) Initially, the probability of selecting any of the channels, $p_i^j(0)$, is set to $1/N$.

2) Select a channel according to the probability distribution, $p_i^j(k)$. Transmit packets during the transmission interval.

3) Based on the measured feedback, update $J_i^j(n)$, $L_i^j(k)$ and $e_i^j(k)$.

4) If $L_i^j(k) \geq M$, update $\hat{H}_i^j(k)$, $\hat{E}_i^j(k)$ and $\hat{\phi}_i^j(k)$ and continue on step 5. Otherwise, go to step 7.

5) $\beta_i^j(k) = \begin{cases} 0, \text{ if } \dfrac{\phi^* - \hat{\phi}_i^j(k)}{\phi^*} < \delta \\ \qquad (\textit{satisfactory response}) \\ 1 \quad \text{otherwise} \\ \qquad (\textit{unsatisfactory response}) \end{cases}$

6) Detect the channel index, $\hat{m}_i$, that provides the best estimated performance, $\hat{\phi}_i^j(k)$.

Update the probabilities if the environmental response was satisfactory, i.e. if $\beta_i^j(k) = 0$,

$$\begin{cases} p_i^{\hat{m}_i}(k+1) = 1 - \sum_{q=1, q \neq \hat{m}_i}^{N} p_i^q(k+1) \\ p_i^l(k+1) = \max(p_i^l(k) - \theta(k), \eta) \quad \forall l \neq \hat{m}_i \end{cases} \tag{2}$$

where $\theta(k) = \begin{cases} \gamma \cdot |\Delta(k)| / \phi^*, & \text{if } -\delta < \Delta(k) / \phi^* \\ \lambda \cdot |\Delta(k)| / \phi^* & \text{otherwise} \end{cases}$, such that $0 \leq \theta(k) < 1$ and $\Delta(k) = \phi^* - \hat{\phi}_i^j(k)$,

where η[1], the minimum possible probability of selecting a channel is chosen such that guarantees all the channels be selected for a minimum certain number of times, K_I, during a certain number of iterations, M_I. This would keep the estimated channel performance values up-to-date.

7) Continue to the next iteration, step 2.

B.2. The Adaptive Pursuit Reward-Penalty Algorithm

The steps in the Pursuit Reward-Penalty learning algorithm are the same as the steps in the Pursuit Reward-Inaction, except for Step 6 – the update law. In this step, when the environmental response is not satisfactory, the probability of selecting the current channel is reduced, and the probability of the other channels are increased as follows.

6) Detect the channel index, $\hat{m}_i$, that provides the best estimated performance, $\hat{\phi}_i^j(k)$.

[1] The minimum probability of selecting a channel is determined such that it satisfies the non-equality below.
Pr{channel *i* being selected at least K_I times over M_I iterations} $\geq \rho$.
This implies that

$$\sum_{j=K_I}^{M_I} C(M_I.j).\eta^j.(1-\eta)^{M_I - j} \geq \rho,$$

If the environmental response was satisfactory, i.e. $\beta_i^j(k) = 0$, $\begin{cases} p_i^{\hat{m}_i}(k+1) = 1 - \sum_{q=1, q\neq \hat{m}_i}^{N} p_i^q(k+1) \\ p_i^l(k+1) = \max(p_i^l(k) - \theta(k), \eta) \quad \forall l \neq \hat{m}_i \end{cases}$

If the environmental response was unsatisfactory, i.e. $\beta_i^j(k) = 1$, and $j \neq \hat{m}_i$,

$$\begin{cases} p_i^j(k+1) = \max(1 - \sum_{q=1, q\neq j}^{N} p_i^q(k+1), \eta) \\ p_i^l(k+1) = p_i^l(k) + \theta(k) / {N-1} \quad \forall l \neq j \end{cases}$$

B.3. The Adaptive Pursuit Reward-Only Algorithm

The steps in the Pursuit Reward-Penalty learning algorithm are the same as the steps in the Pursuit Reward-Inaction, except for Step 6 – the update law. In this scheme, the probabilities are updated such that selecting the channel with the highest performance is “pursued.” This update is performed regardless of the “satisfactory” or “unsatisfactory” condition of the performance. Anyhow, we want to increase the probability of selecting the channel which provides the highest performance – even if this performance is less that the desired performance. However, the magnitude of the update step is determined by the relative error of the performance, $|\Delta(k)| / \phi^*$. The update law, Step 6, of the algorithm is as follows.

6) Detect the channel index, $\hat{m}_i$, that provides the best estimated performance, $\hat{\phi}_i^j(k)$.

Update the probabilities regardless of the environmental response. The probability of selecting the channel that provides the highest performance is increased and the probabilities of the other channels are reduced as following.

$$\begin{cases} p_i^{\hat{m}_i}(k+1) = 1 - \sum_{q=1, q\neq \hat{m}_i}^{N} p_i^q(k+1) \\ p_i^l(k+1) = \max(p_i^l(k) - \theta(k), \eta) \quad \forall l \neq \hat{m}_i \end{cases}$$

where $M_I \geq K_I.N$ (N is the number of available channels).

IV. Simulation Results and Discussions

In this section, we present the numerical results of running the three learning algorithm on a set of peer-to-peer wireless networks with varying traffic, mobility, and number of nodes. The simulations were performed using network simulator NS-2. Moreover, it was modified to implement the three learning algorithms: Adaptive PRI, Adaptive PRP, and Adaptive PRO. The networks are consisted of 50 single-radio wireless nodes located in an area of 100m×100m, while the communication range of the nodes are 250m. As a result, a dense network topology is created where a single channel is not able to provide sufficient quality of service (QoS). Traffic is generated by a constant bit rate (CBR) sources with data rates equal to 2 Mbps and packet size equal to 1024 bytes. The simulations considered networks with up to 11 orthogonal channels whose bandwidth is set to 11 Mbps. The objective of the multi-channel protocol is to allocate the available channels to the links such that the performance converges to a desired value as defined in Equation (1). The target value ϕ^* and the update parameters were set for different scenarios such that the desired performance is achievable. The nodes start without preferred channel and switch between channels until they find the one that provides the desired performance. The width of the moving average window, M, was selected to be 5.

A. Static Scenario – starting flows at different times

This simulation scenario considers single time slot duration, T_s, where all nodes are contending for the channels. The three Adaptive learning algorithms were run on the networks of 50 nodes with up to 11 orthogonal channels. Three flows start at second 2, then seven more flows start at second 3 and finally fifteen more flows start at second four. The standard 802.11 protocol was also run on the networks to compare its performance to the performance of the learning algorithms. This was done by *a)* using a single channel, and *b)* using 10 channels and randomly allocating them to the links. For each case, the simulation was repeated using 10 random scenarios, and the average of the 10 repeated simulations were used in result analysis. The achieved throughput by applying the different methods is presented in Table 1.

It is noticed that as the number of channels used in the Adaptive PRI learning schemes is increased, the throughput is significantly increased compared to the single-channel 802.11 scenario. The increased throughput is provided by the additional capacity of the additional channels. Naturally when there are only 3 flows in the network, we do not expect the throughput to improve by increasing the number of channels to higher than 3. For the case of 25 flows, the Adaptive PRI with 10 data channels provides an improvement of 13 times in throughput compared to a single-channel 802.11. When there are 25 flows in the network and only one channel is provided, the network is so congested that it provides a throughput of only 3 for the 25 flows. However, when the Adaptive PRI is used on 10 channels, it provides a higher capacity though not the capacity required to eliminate the congestion. The capacity provided by the 10 channels is almost 10×capacity of each channel. The capacity of each channel for data packets in 802.11 is almost half of the channel bandwidth. We had chosen a standard channel bandwidth of 11Mbps in the simulations. Therefore the total throughput of 39.58 Mbps is reasonable compared to the total capacity of almost 50 Mbps, since there is a noticeable congestion in the network. Also for the same case of 25 flows, PRI with 10 data channels provides an improvement of 1.22 times in throughput over random allocation of 10 channels. Using the Adaptive PRI algorithm for the networks of 6 nodes and 20 nodes, the maximum possible throughput (6 Mbps and 20 Mbps, respectively) can be achieved by utilizing 3 and 10 channels respectively, which will allocate a different channel to each link. However, for the network of 50 nodes saturation and high drop rate are inevitable, although the throughput is improved significantly by increasing the number of channels. As the number of nodes in the network increase, the number of contending nodes during the time slot, T_s, and mini slot, T_{mini}, increases. This can result in a case that some nodes do not get any chance to transmit during T_{mini}. Hence with a performance much smaller than the desired performance (i.e., unsatisfactory environment response), due to the “reward” characteristic of the learning algorithm, probabilities of channel selection would not be updated for them. We will get back to this issue later.

Table 1 also presents the drop rate in the network using the different methods of channel allocations, and different number of channels. The results show that for the networks of 3 and 10 flows, the drop rate is significantly reduced by utilizing the Adaptive PRI learning scheme and more number of channels. The drop rate for the network of 25 flows is also reduced, but not as much as it was for the networks with smaller densities. This is due to the fact that the network is so dense and the number of contending nodes is so high that the saturation is inevitable. It can be noticed by using the Adaptive PRI channel allocation and 10 data channels, in the worst case scenario (greatest number of flows), the drop rate is reduced by 78.38% compared to when using a single-channel 802.11. For the same case of 25 flows, PRI with 10 data channels provides a 44.78% reduction on drop rate over random allocation of 10 channels.

Table 1 presents the energy consumption per packet in the network using the different methods of channel allocations, and different number of channels. The results show that using the PRI learning scheme and increasing the number of data channels significantly improves the energy consumption per packet. It can be noticed that by using PRI channel allocation and 10 data channels, in the worst case scenario (greatest number of flows), the energy consumption is reduced by 90.25% compared to when using a single-channel 802.11. Also using PRI with data channels reduces the energy consumption by 12.33%. For the same case of 25 flows, PRI with 10 data channels provides a 12.33% reduction in energy consumption per packet over random allocation of 10 channels.

Another performance metric that was used for evaluating the channel allocation schemes was fairness index [16]. Table 1 also presents the fairness index provided by using the different methods of channel allocations, and different number of channels. The results show that using the Adaptive PRI learning scheme and increasing the number of data channels improves the fairness index – especially when there are greater number of flows. It can be noticed that by using the Adaptive PRI channel allocation and 10 data channels, in the worst case scenario (greatest number of flows), the fairness index is increased by 3.7 times compared to when using a single-channel 802.11. Also using the Adaptive PRI with 10 data channels increases the fairness index by 1.28%. For the same case of 25 flows, the Adaptive PRI with 10 data channels provides a 1.28% improvement in fairness over random allocation of 10 channels.

The other two channel allocation learning schemes, i.e. Adaptive PRP and Adaptive PRO, were also applied to the same networks and scenarios, with 10 data channels. Table 2 shows the throughput over the network when using the Adaptive PRI, PRP and PRO schemes and 10 data channels. It is noticed that for the greater number of flows, the Adaptive PRP schemes provides a slightly higher throughput compared to the other two learning schemes. Table 2 also shows the drop rate over the network when using the Adaptive PRI, PRP and PRO schemes and 10 data channels. It is noticed that for the greater number of flows, the PRI scheme provides a slightly higher (worse) drop rate compared to the other two learning schemes.

Table 2 shows the energy consumption per packet in the network when using the Adaptive PRI, PRP and PRO schemes and 10 data channels. The three methods do not show any significant difference in the sense of energy consumption. The fairness index of the network, when using the Adaptive PRI, PRP and PRO schemes and 10 data channels, is shown in Table 2. It is noticed that for the greater number of flows, the Adaptive PRP scheme provides a slightly higher (better) fairness compared to the other two learning schemes.

We also examined a case in which all the 25 flows started at second 2, then they were reduced to 10 flows at second 3, and finally reduced to 3 flows at second 4. Similarly the simulations were performed for 10 random scenarios for a network of 10 data channels, using the Adaptive PRI learning automata scheme. By comparing Table 3 to Table 1, it can be concluded that by starting a greater number of flows at the same time, a smaller throughput can be achieved. That is, when 25 flows start at the same time, the achieved throughput is limited to 36.76 Mbps (Table 3), while by adding 15 flows to the previously existing 19 flows (Table 1) a throughput of 39.58 Mbps can be achieved. The reason for the smaller achieved throughput is the high collision in the case of the simultaneously starting greater number of flows.

B. Mobile Scenario

In Section IV.A (static scenario) we mentioned the assumption of a static network topology during $T_{S.}$ In this section we examine a case that the network topology undergoes changes during the T_s period. We consider a larger network (1000mx1000m) and greater number of flows (50 flows, i.e. 100 peer-to-peer nodes). Then the behavior of the single-channel 802.11, randomly allocated 10 channels using 802.11, and the Adaptive PRI learning scheme in the case of mobility of the nodes were examined. For four different values of maximum speed (5, 10, 15, and 20 m/s) and also static case (0 m/s), 10 random scenarios were generated and the average of these repeated simulations were used for comparison. Table 4 presents the results for using the Adaptive PRI and 10 channels. The speed change does not show a significant effect on the performance. However, in general, these larger network scenarios with a higher traffic flow show a lower performance compared to the static case (Section IV.A).

By using the Adaptive PRI learning scheme, the throughput, drop rate, energy consumption and fairness index show a significant improvement compared to the case that 802.11 is used with randomly allocated 10 data channels (Table 4). The throughput is improved by 19.6%, the drop rate is reduced by 47.6%, the energy consumption per packet is reduced by 10.6% and the fairness index is improved by 11.4%. Also compared to the single-channel 802.11, both Adaptive PRI and 802.11 over randomly allocated 10-data channel are performing significantly better.

C. Comparison of the three schemes of the learning automata regarding to probability update

Earlier we mentioned the problem of ‘left-out’ links in the PRI algorithm. This problem occurs when none of the channels provide a satisfactory performance, and hence the probabilities of channel selections are not updated at all. This case is examined below, where the Adaptive PRI is used for channel allocation in a peer-to-peer network of 50 nodes (25 links) using 10 channels.

It was observed that the channel allocations of 21 links out of 25 links converged. The channel allocations for the links 7, 9, 22, and 23 always provided a performance much smaller than the desired performance (i.e., unsatisfactory environment response). Due to the “reward-inaction” characteristic of the learning algorithm, probabilities of channel selection for these links would not be updated. These links are ‘left-out’ of the update process. The probabilities of channel selections for one of the converged links (link 15), and one of the non-converged links (link 7) are shown in Figure 2 and Figure 3, respectively. Figure 2 shows how the probabilities of selecting the channels converge for link 15 while Figure 3 shows that these probabilities are not updated at all. All the channels keep their initial equal probability, 0.1, all the time. In each iteration one of the channels is selected randomly.

By using the Pursuit Reward-Penalty algorithm, the 'left-out' links problem is eliminated and the probability of selecting the channels is updated even if the channel allocation is not providing a satisfactory performance. Although the probabilities of channel selections are updated, the channel allocations for 6 links (links 5, 7, 12, 21, 22, and 23) do not converge yet by the end of the simulation. The channel allocations for the mentioned links provide a performance much smaller than the desired performance (i.e., unsatisfactory environment response). Hence the probabilities of channel selection for these links are updated through the "penalty" process of the algorithm. The probabilities of channel selections for one of the converged links (link 15), and one of the yet non-converged links (link 7) are shown in Figure 4 and Figure 5, respectively. Figure 4 shows how the probabilities of selecting the channels converges for link 15, and Figure 5 shows that these probabilities for link 7 are converging, though slowly (parameter adjustment might be needed or increasing the speed here).

Figure 6 shows the changes in the channel allocations as the Pursuit Reward-Only algorithm runs on the network. It shows that the channel allocations of all the links converge. The probabilities of channel selection for all the links are updated with the "pursuit" characteristic regardless of the environment response (channel performance). The updates are performed such that the probability of selecting the channel with the best performance is increased, and the probabilities of selecting the other channels are decreased. The magnitude of the relative error determines the magnitude of the update step.

Comparison of the results of the three algorithms shows that the Pursuit Reward-Penalty provides update and convergence for the cases that the channel performance is significantly smaller than the desired performance. The Pursuit Reward-Inaction did not guarantee the update for the less than desirable performance. This would result in “left-out” links; the links with no converged channel allocation. On the other hand, the Pursuit Reward-Only algorithm always increases the probability of the channel with the highest performance, whether the performance of the selected channel is satisfactory or not. This algorithm provides the fastest convergence among the three algorithms.

V. Conclusions

In this paper we propose a distributed dynamic channel allocation algorithm for wireless networks whose nodes are equipped with single radio interface. We make the single-radio assumption for the sake of simplicity of the network, planning to apply the learning algorithm to wireless ad-hoc sensor networks. The periodic nature of the algorithm makes it dynamic and enables the channel allocation to adapt to the topographic changes, possible loss of some channels, mobility of the nodes, and the traffic flow changes. The Adaptive Pursuit learning algorithm runs periodically on the nodes, and adaptively finds the optimum channel allocation that provides the desired performance. By selecting realistic desired performance metric, the convergence of the algorithm can be guaranteed. The analytical proof of convergence is presented in this appendix, and also the simulation results for networks of different densities and data channels were provided and showed a significant improvement in throughput, drop rate, energy consumption per packet, fairness index when compared to the single-channel. 802.11 and random allocation of the channels.

Also in order to avoid the ‘left-out’ links in the learning process in the first algorithm (Adaptive PRI), we proposed using the other two algorithms, Pursuit Reward-Penalty and Pursuit Reward-Only algorithms.

We compared the results of these two algorithms to the results of the Pursuit Reward-Inaction, and showed that the Pursuit Reward-Penalty eliminates the ‘left-out’ links problem, and provides convergence using the same parameters as used in the Pursuit Reward-Inaction. The Pursuit-Only algorithm also eliminates the ‘left-out’ links problem. Also with the same parameters, it provides a faster convergence compared to the Pursuit Reward-Penalty algorithm.

VI. Acknowledgements

The authors acknowledge the support of the Intelligent Systems Center and Air Force Research Lab.

Appendix A. Proof of Convergence

In Section II.B, the channel allocation algorithms were presented. In this section, the proofs of convergence of the algorithms are presented. The proofs follow the general method used in [14].

A. Proof of Convergence of the Adaptive Pursuit Reward-Inaction Algorithm

Theorem I establishes that for each node that is running the algorithm, if after a certain time, the channel allocation results in a greater performance for one channel compared to the other channels, the probability of selecting that channel tends to 1. Theorem II establishes that for each node and each channel, there exists a time that the channel has been selected by the node for at least *M* times. This guarantees having the average throughput, delay and consumed energy values, which are required for the performance evaluation.

<u>Theorem I:</u> Suppose there exists an index m_i and a time instant $k_0 < \infty$ such that $\hat{\phi}_i^{m_i}(k) > \hat{\phi}_i^{j}(k)$ for all j such that $j \neq m_i$ and all $k \geq k_0$. Then there exists γ_0 and λ_0 such that for all resolution parameters $(\gamma < \gamma_0, \lambda < \lambda_0)$, $p_i^{m_i}(k) \to 1$ with probability 1 as $k \to \infty$.

<u>Proof:</u> From the definition for Discrete Pursuit Reward-Inaction, we know that if m_i satisfies

$m_i = \arg \max_j \hat{\phi}_i^{j}(k)$, where $\hat{\phi}_i^{m_i}(k) = \max_j \hat{\phi}_i^{j}(k)$, then $\hat{\phi}_i^{m_i}(k) > \hat{\phi}_i^{j}(k)$ for all $j \neq m_i$ and all $k \geq k_0$.

Therefore, for all $k > k_0$, $$p_i^{m_i}(k+1) = \begin{cases} 1 - \sum_{j=1,\, j\neq m_i}^{N} (p_i^j(k) - \theta(k)), \\ \qquad \text{if } \beta_i^l(k) = 0 \quad (\text{w.p. } \zeta_i^{m_i}(k)) \\ p_i^{m_i}(k) \\ \qquad \text{if } \beta_i^l(k) = 1 \quad (\text{w.p. } 1 - \zeta_i^{m_i}(k)) \end{cases}$$

If $p_i^{m_i}(k) = 1$, then the "pursuit" property of the algorithm trivially proves the result.

Assuming that the algorithm has not yet converged to the m_ith channel, there exists at least one nonzero component of $\mathbf{P}_i(k)$, $p_i^q(k)$, with $q \neq m_i$. Therefore we can write

$p_i^q(k+1) = p_i^q(k) - \theta(k) < p_i^q(k)$.

Since $\mathbf{P}_i(k)$ is a probability vector, $\sum_{j=1}^{N} p_i^j(k) = 1$, and $p_i^{m_i}(k) = 1 - \sum_{j=1, j\neq m_i}^{N} p_i^j(k)$. Therefore,

$1 - \sum_{j=1, j\neq m_i}^{N} (p_i^j(k) - \theta(k)) > p_i^{m_i}(k)$.

As long as there is at least one nonzero component, $p_i^q(k)$ (where $q \neq m_i$), it is clear that we can decrement $p_i^q(k)$ and increment $p_i^{m_i}(k)$ by at least $\theta(k)$. Hence, $p_i^{m_i}(k+1) = p_i^{m_i}(k) + c(k)\cdot\theta(k)$, where $c(k)\cdot\theta(k)$ is an integral multiple of $\theta(k)$, and $0 < c(k) < N$, and

$$\theta(k) = \begin{cases} \gamma \cdot |\Delta(k)| / \phi^*, & \text{if } -\delta < \Delta(k)/\phi^* \\ \lambda \cdot |\Delta(k)| / \phi^* & \text{otherwise} \end{cases}$$

Therefore we can express the expected value of $p_i^{m_i}(k+1)$ conditioned on the current state of the channel, $\mathbf{Q}(k)$, $(\mathbf{Q}(k) = \mathbf{P}_i(k), \varphi_i(k))$ as follows

$$E[p_i^{m_i}(k+1) \mid \mathbf{Q}(k), p_i^{m_i}(k) \neq 1] = \zeta_i^{m_i}(k)\cdot[p_i^{m_i}(k) + c(k)\cdot\theta(k)] + 1 - \zeta_i^{m_i}(k) \cdot p_i^{m_i}(k)$$
$$\triangleright \qquad = p_i^{m_i}(k) + \zeta_i^{m_i}(k)\cdot c(k)\cdot\theta(k)$$

Since all the previous terms have an upperbound of unity, $E[p_i^{m_i}(k+1) \mid \mathbf{Q}(k), p_i^{m_i}(k) \neq 1]$ is also bounded,

$\sup_{k\geq 0} E[p_i^{m_i}(k+1) \mid \mathbf{Q}(k), p_i^{m_i}(k) \neq 1] < \infty$.

Thus we can write $E[p_i^{m_i}(k+1) - p_i^{m_i}(k) \mid \mathbf{Q}(k)] = \zeta_i^{m_i}(k)\cdot c(k)\cdot\theta(k) \geq 0$, for all $k \geq k_0$

implying that $p_i^{m_i}(k)$ is submartingale. By submartingale convergence theorem, the sequence $\{p_i^{m_i}(k)\}_{k\geq k_0}$ converges.

Therefore $E[p_i^{m_i}(k+1) - p_i^{m_i}(k) \mid \mathbf{Q}(k)] \to 0$ w.p.1, as $k \to \infty$.

This implies that $\zeta_i^{m_i}(k)\cdot c(k)\cdot\theta(k) \to 0$ w.p.1. This in turn implies that $c(k) \to 0$ w.p.1 $(\theta(k) \to 0$ w.p.1$)$, which means there is no nonzero element in $\mathbf{P}_i(k)$ except for $p_i^{m_i}(k)$ (or

$\Delta(k) \to 0$). Consequently, $\sum_{j=1, j\neq m_i}^{N} p_i^j(k) \to 0$ w.p.1 and $p_i^{m_i}(k) = 1 - \sum_{j=1, j\neq m_i}^{N} p_i^j(k) \to 1$ w.p.1

■

Theorem II: For each node *i* and channel *j*, assume $p_i^j(0) \neq 0$. Then for any given constant $\delta_0 > 0$ and $M < \infty$, there exists $\gamma_0 < \infty$, $\lambda_0 < \infty$ and $k_0 < \infty$ such that under the Discrete Pursuit Reward-Inaction algorithm, for all learning parameters $\gamma < \gamma_0$ and $\lambda < \lambda_0$ and all time $k > k_0$:

Pr{each channel chosen by node *i* more than *M* times at time *k***}** $\geq 1 - \delta_0$.

Proof: Define the random variable $Y_i^j(k)$ as the number of times that channel *j* was chosen by node *i* up to time *k*. then we must prove that $\Pr\{Y_i^j(k) > M\} \geq 1 - \delta_0$. This is equivalent to proving

$$\Pr\{Y_i^j(k) \leq M\} \leq \delta_0 . \tag{A.1}$$

The events $Y_i^j(k) = q$ and $Y_i^j(k) = s$ are mutually exclusive for $q \neq s$, so we can rewrite Equation (A.1) as

$$\sum_{q=1}^{M} \Pr\{Y_i^j(k) = q\} \leq \delta_0 .$$

For any iteration of the algorithm, **Pr{**choosing channel *j***}** ≤ 1. Also the magnitude by which any channel selection probability can decrease in any iteration is bounded by $\gamma \cdot |\Delta(k)| / \phi^*$ (or $\lambda \cdot |\Delta(k)| / \phi^*$), where $\Delta(k) < \Delta$ for all *k*. During any of the first *k* iterations of the algorithm:

$$\Pr\{\text{channel } j \text{ is not chosen by node } i\} \leq \left(1 - p_i^j(0) + k \cdot \gamma \cdot |\Delta| / \phi^*\right).$$

Using these upper bounds, the probability that channel *j* is chosen at most *M* times among *k* choices, has the following upper bound

$$\Pr\{Y_i^j(k) \leq M\} \leq \sum_{l=1}^{M} C(k,l)(1)^l (1 - p_i^j(0) + k \cdot \gamma \cdot |\Delta| / \phi^*)^{k-l} \tag{A.2}$$

In order to make a sum of *M* terms less than δ_0, it is sufficient to make each term less than δ_0 / M. Consider an arbitrary term, $l = m$. We must show that

$C(k,m)(1)^m(1-p_i^j(0)+k\cdot\gamma\cdot{|\Delta|}/{\phi^*})^{k-m} < {\delta_0}/{M}$, or $M\cdot C(k,m)(1)^m(1-p_i^j(0)+k\cdot\gamma\cdot{|\Delta|}/{\phi^*})^{k-m} < \delta_0$.

Knowing that $C(k,m)\le k^m$, we have to prove that $M\cdot k^m\left(1-p_i^j(0)+k\cdot\gamma\cdot{|\Delta|}/{\phi^*}\right)^{k-m}\le\delta_0$.

Now in order to get the L.H.S of this term to be less than δ_0 as k increases, $\left(1-p_i^j(0)+k\cdot\gamma\cdot{|\Delta|}/{\phi^*}\right)$ must be strictly less than unity. In order to guarantee this, we bound the value of γ with respect to k in such a way that $\left(1-p_i^j(0)+k\cdot\gamma\cdot{|\Delta|}/{\phi^*}\right)<1$. We can achieve this by requiring that $\gamma<\frac{p_i^j(0)}{k\cdot|\Delta|}\cdot\phi^*$.

Let $\gamma=\frac{p_i^j(0)}{2k\cdot|\Delta|}\cdot\phi^*$. (A.3)

With this value of γ, Equation (A.2) is simplified to $\Pr\{Y_i^j(k)\le M\}<M\cdot k^m\cdot\psi^{k-m}$, where $\psi=1-\frac{1}{2}p_i^j(0)$, and $0<\psi<1$. Now we need to evaluate $\lim_{k\to\infty} M\cdot k^m\cdot\psi^{k-m}$.

$$\lim_{k\to\infty} M\cdot k^m\cdot\psi^{k-m}=M\cdot\lim_{k\to\infty}\frac{k^m}{\left({1}/{\psi}\right)^{k-m}}\text{ , with } \gamma=\frac{p_i^j(0)}{2k\cdot|\Delta|}\cdot\phi^*.$$

By applying l’Hopital’s rule:

$$\begin{aligned} M\cdot\lim_{k\to\infty}\frac{k^m}{\left({1}/{\psi}\right)^{k-m}} &= M\cdot\lim_{k\to\infty}\frac{m!}{\left(\ln({1}/{\psi})\right)^m\left({1}/{\psi}\right)^{k-m}} \\ &= 0 \qquad\qquad \text{with } \gamma=\frac{p_i^j(0)}{2k\cdot|\Delta|}\cdot\phi^* \end{aligned}$$

Therefore Equation (A.2) has a limit of zero as $k\to\infty$ and $\gamma\to 0$, whenever Equation (A.3) is satisfied.

Since the limit exists, for every channel j there is a $k(j)$ such that for all $k>k(j)$, Equation (A.2) holds.

Now set $\gamma(j)=\frac{p_i^j(0)}{2k(j)\cdot|\Delta|}\cdot\phi^*$. It remains to be shown that Equation (A.2) is satisfied for all $\gamma<\gamma(j)$, and for all $k>k(j)$. This is trivial because as γ decreases, the L.H.S of Equation (A.2) is monotonically decreasing, and so the inequality (A.2) is preserved.

Also for any $k > k(j)$, since $Y_i^j(k(j)) \geq M \Rightarrow Y_i^j(k) \geq M$, by the laws of probability:

$$\Pr\{Y_i^j(k) \geq M\} \geq \Pr\{Y_i^j(k(j)) \geq M\}.$$

Thus in this case also, the inequality (A.2) still holds. Hence for any channel *j*, $\Pr\{Y_i^j(k) \leq M\} \leq \delta_0$ whenever $k > k(j)$ and $\gamma < \gamma(j)$. Since we can repeat this argument for all the channels, we can define k_0 and γ_0 as $k_0 = \max_{1 \leq j \leq N} \{k(j)\}$, and $\gamma_0 = \max_{1 \leq j \leq N} \{\gamma(j)\}$. Thus for all *j*, it is true that for all $k > k_0$ and $\gamma < \gamma_0$ ($\lambda < \lambda_0$), the quantity $\Pr\{Y_i^j(k) \leq M\} \leq \delta_0$ and theorem is proved. ■

References

[1] Joseph Mitola III. *Cognitive Radio An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2000.

[2] Bahl, P., Chandra, R., and Dunagan, J., "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," In *Proceedings of the 10th Annual international Conference on Mobile Computing and Networking* (Philadelphia, PA, USA, September 26 - October 01, 2004). MobiCom '04. ACM, New York, NY, pp:216-230.

[3] Alicherry, M., Bhatia, R., and Li, L. 2005. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proceedings of the 11th Annual international Conference on Mobile Computing and Networking* (Cologne, Germany, August 28 - September 02, 2005). MobiCom '05. ACM, New York, NY, pp.58-72.

[4] Mishra, A., Banerjee, S., and Arbaugh, W., "Weighted coloring based channel assignment for WLANs," *SIGMOBILE Mob. Comput. Commun. Rev.* 9, 3 (Jul. 2005), pp:19-31.

[5] Nie, N.; and Comaniciu, C., "Adaptive channel allocation spectrum etiquette for cognitive radio networks," *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on* , vol., no., pp.269-278, 8-11 Nov. 2005.

[6] Li, Jiandong; Chen, Dong; Li, Weiying; Ma, Jing, "Multiuser Power and Channel Allocation Algorithm in Cognitive Radio," *Parallel Processing, 2007. ICPP 2007. International Conference on* , vol., no., pp.72-72, 10-14 Sept. 2007.

[7] A. Raniwala, K. Gopalan and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh network," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp: 50-65, Apr. 2004.

[8] Felegyhazi, M.; Cagalj, M.; Bidokhti, S. S.; and Hubaux, J.-P., "Non-Cooperative Multi-Radio Channel Allocation in Wireless Networks," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* , vol., no., pp.1442-1450, May 2007.

[9] P. Kyasanur and N. H. Vaidya, "Routing in Multi-Channel Multi-Interface Ad Hoc Wireless Networks," technical report, Dec. 2004.

[10] Kyasanur, P.; and Vaidya, N.H., "Routing and interface assignment in multi-channel multi-interface wireless networks," Proc. of *Wireless Communications and Networking Conference*, vol.4, no., pp. 2051-2056 Vol. 4, 13-17 March 2005.

[11] Jungmin So and Nitin Vaidya, "Routing and channel assignment in multi-channel multi-hop wireless networks with single-NIC devices," *Technical Report, University of Illinois at Urbana Champaign*, Dec. 2004.

[12] Han, Z.; Ji, Z.; and Liu, K.J.R., "Fair Multiuser Channel Allocation for OFDMA Networks Using Nash Bargaining Solutions and Coalitions," *Communications, IEEE Transactions on* , vol.53, no.8, pp. 1366-1376, Aug. 2005.

[13] Patel, J. A.; Luo, H.; and Gupta, I., "A Cross-Layer Architecture to Exploit Multi-Channel Diversity with a Single Transceiver," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* , vol., no., pp.2261-2265, May 2007.
[14] Oommen, B.J.; and Lanctot, J.K., "Discretized pursuit learning automata," *Systems, Man and Cybernetics, IEEE Transactions on* , vol.20, no.4, pp.931-938, Jul/Aug 1990.
[15] Haleem, M.A.; and Chandramouli, R., "Adaptive downlink scheduling and rate selection: a cross-layer design," *Selected Areas in Communications, IEEE Journal on* , vol.23, no.6, pp. 1287-1297, June 2005.
[16] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems", DEC Research Report TR-301, September 1984.
[17] T. Clausen, P. Jacquet, "Optimized Link State protocol (OLSR)," 2003.
[18] N. Regatte and S. Jagannathan "Optimized Energy-Delay Routing in Ad Hoc Wireless Networks," *Proc. of the World Wireless Congress*, May 2005.

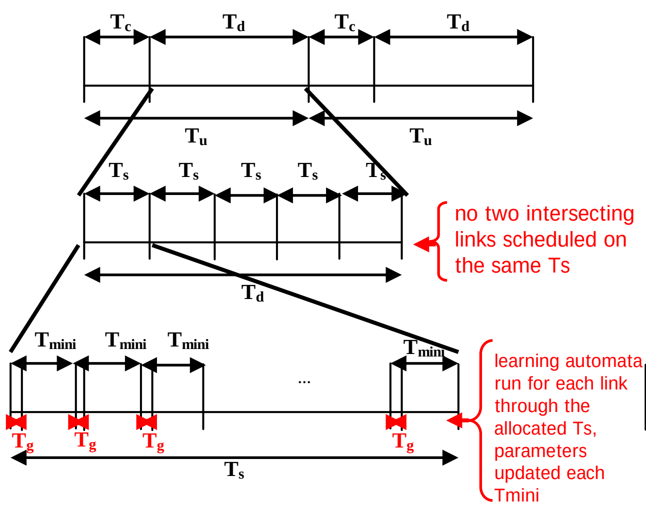

Figure 1. The two periods of control and data, and time slots within the data transmission period.

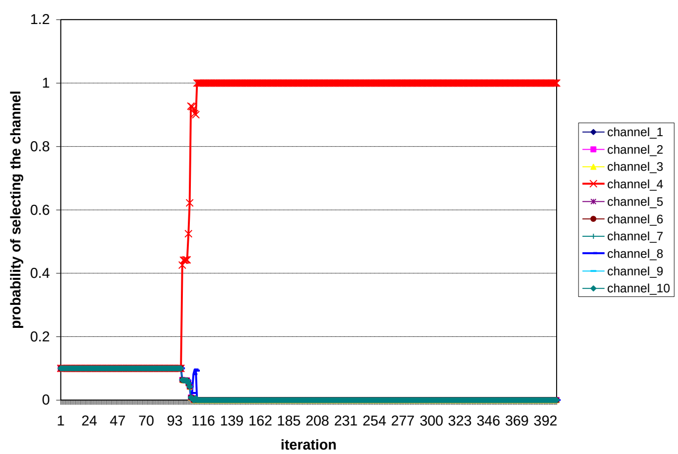

Figure 2. The probability of selecting the channels for link_15, using the Adaptive Pursuit Reward-Inaction algorithm.

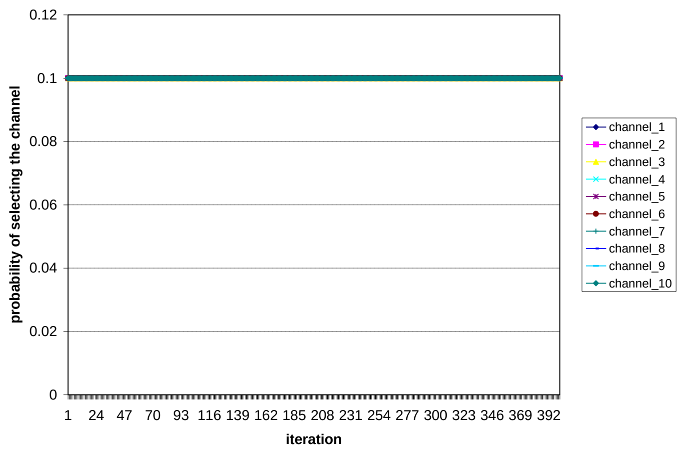

Figure 3. The probability of selecting the channels for link_7, using the Pursuit Reward-Inaction algorithm.

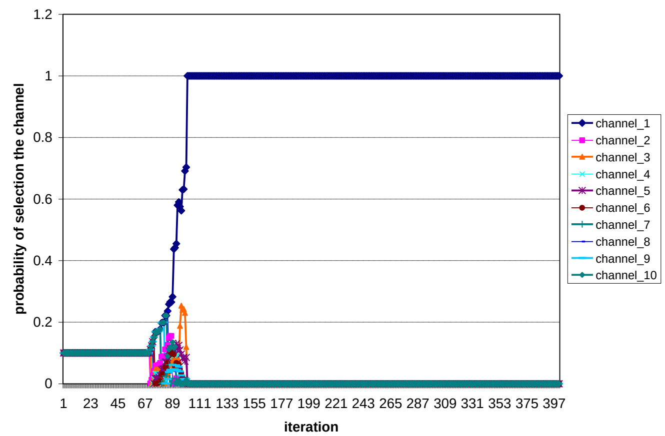

Figure 4. The probability of selecting the channels for link_15, using the Pursuit Reward-Penalty algorithm.

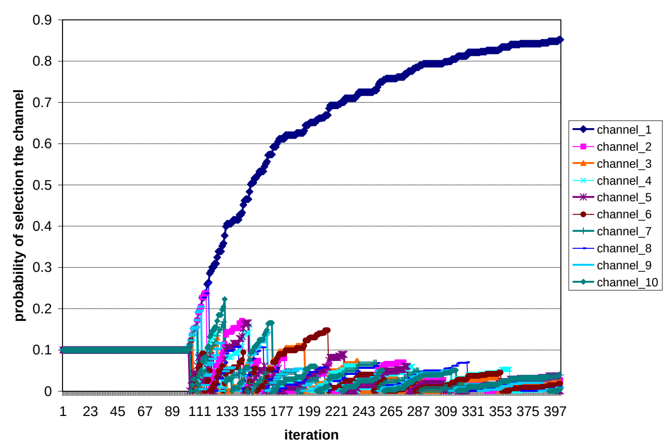

Figure 5. The probability of selecting the channels for link_7, using the Pursuit Reward-Penalty algorithm.

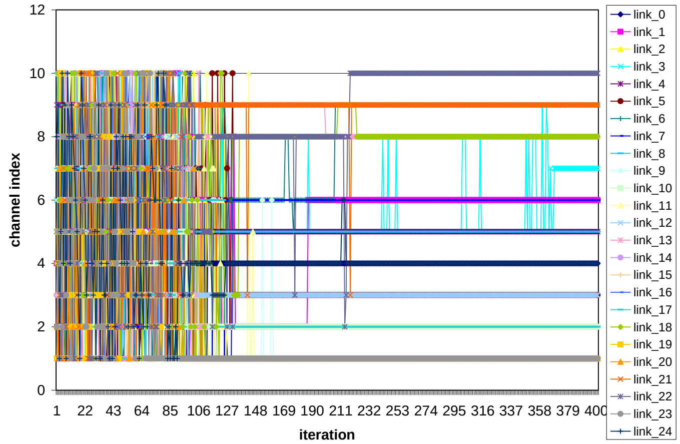

Figure 6. Channel allocation for 25 links in a network of 50 peer-to-peer nodes, using Pursuit Reward-Only learning automata. Channel allocations for all the links have converged.

Table 1. Throughput, drop rate, energy consumption, and fairness index of the network with different channel allocation schemes.

	Throughput (Mbps)			Drop rate(Mbps)			Energy consumption (joules/packet)			Fairness index		
	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows
802.11 – single data channel	4.20	3.89	3.00	0.77	15.98	47.00	0.00215	0.00807	0.01969	0.8028	0.4443	0.2157
Adaptive PRI – 2 data channels	6.15	8.25	7.83	0	11.75	42.94	0.00140	0.00331	0.00774	0.9620	0.7344	0.4082
Adaptive PRI – 3 data channels	6.12	12.44	12.19	0	5.82	38.80	0.00125	0.00235	0.00521	0.9716	0.8337	0.5129
Adaptive PRI – 6 data channels	6.10	19.35	24.80	0	0.26	26.13	0.00114	0.00153	0.00284	0.9789	0.9090	0.7244
Adaptive PRI – 8 data channels	6.10	20.34	32.70	0	0	17.76	0.00111	0.00135	0.00226	0.9811	0.9431	0.7689
Adaptive PRI – 10 data channels	6.15	20.57	39.58	0	0	10.16	0.00109	0.00130	0.00192	0.9824	0.9531	0.8022
802.11 – 10 data channels, random channel allocation	6.20	18.80	32.53	0	0.65	18.40	0.00105	0.00142	0.00219	0.9811	0.9475	0.7921

Table 2. Throughput, drop rate, energy consumption, and fairness index when using the three learning schemes of channel allocation and 10 data channels.

	Throughput (Mbps)			Drop rate (Mbps)			Energy consumption (joules/packet)			Fairness index		
	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows
Adaptive PRI – 10 data channels	6.15	20.57	39.58	0	0	10.16	0.001085	0.001301	0.001924	0.9824	0.9531	0.8022
Adaptive PRP– 10 data channels	6.15	20.61	39.95	0	0	9.93	0.001085	0.001312	0.001917	0.9824	0.9507	0.8080
Adaptive PRO – 10 data channels	6.15	20.59	39.82	0	0	9.91	0.001085	0.001301	0.001915	0.9824	0.9527	0.8027

Table 3. Performance metrics of a network with flows. The channel allocation performed using the Adaptive PRI and 10 data channels.

	PRI, 10 data channels		
	25 flows	10 flows	3 flows
Throughput (Mbps)	36.76	25.03	6.14
Drop rate (Mbps)	4.90	0.09	0
Energy consumption (joules/packet)	0.002042	0.001249	0.000991
Fairness index	0.7501	0.9021	0.9832

Table 4. Performance of the Adaptive PRI with 10 data channels on a network of 50 flows, while nodes moving in different speeds. Also performance of the single-channel 802.11and randomly allocated 10 data channels using 802.11 on the same network, while nodes moving at a maximum speed of 10 m/s.

	Adaptive PRI, 10 data channels					802.11 - single channel	802.11 – 10 data channels, randomly allocated
	Static (0 m/s)	5 m/s	10 m/s	15 m/s	20 m/s	10 m/s	10 m/s
Throughput (Mbps)	84.31	83.68	82.96	81.84	79.44	15.51	69.97
Drop rate (Mbps)	13.35	14.10	14.62	15.71	17.78	80.43	26.92
Energy consumption (joules/packet)	0.001734	0.001735	0.001741	0.001760	0.001811	0.008398	0.001940
Fairness index	0.7066	0.6975	0.6900	0.6868	0.6636	0.2169	0.6263